

CAHIER DES CHARGES TECHNIQUES

Projet Borne automatique



Dossier rédiger par Clément LIN



Sommaire

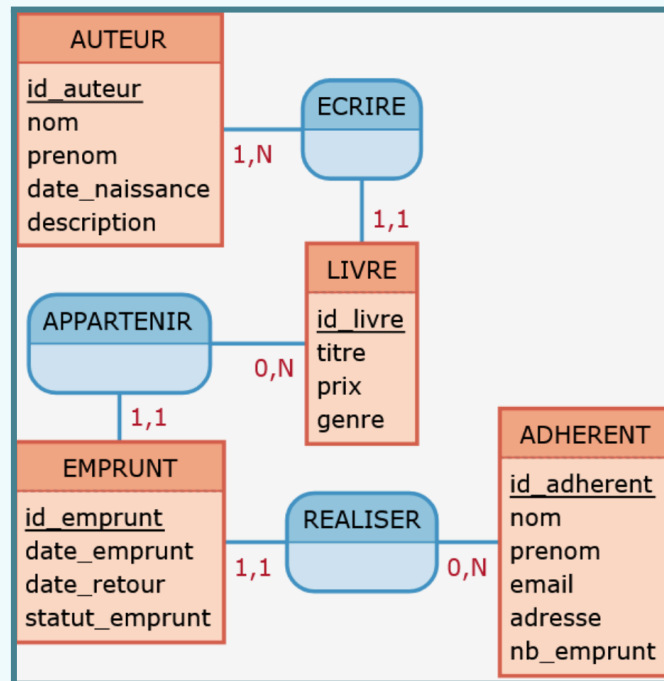
Rédaction de cahier de charge.....	3
1. MCD.....	3
2. MRD.....	3
3. Diagramme de classes.....	4
4. Diagramme de cas.....	4
5. Liste des règles de gestion.....	5
1. Authentification et Accès Utilisateur.....	5
2. Gestion des Emprunts.....	5
3. Réservations et Retours.....	6
4. Gestion des Stocks et Catalogue.....	6
5. Sécurité et Maintenance.....	6
6. Interaction avec le Personnel.....	6
7. Mise à Jour du Logiciel.....	6
6. Diagramme de Gantt (gestion de projet).....	6
7. Description de l'environnement technologique.....	7
7.1. Pare feu.....	7
7.2. Anti-virus.....	7
7.3. OS version.....	7
7.4. IDE Version.....	8
7.5. SGBD.....	8
7.6. Navigateur.....	8
7.7. Github.....	8
7.8. Les type de tests unitaires/fonctionnel.....	8

Rédaction de cahier de charge



1. MCD

Mocodo



2. MRD

Mocodo

ADHERENT (id_adherent, nom, prenom, email)

AUTEUR (id_auteur, nom, prenom, date_naissance , description)

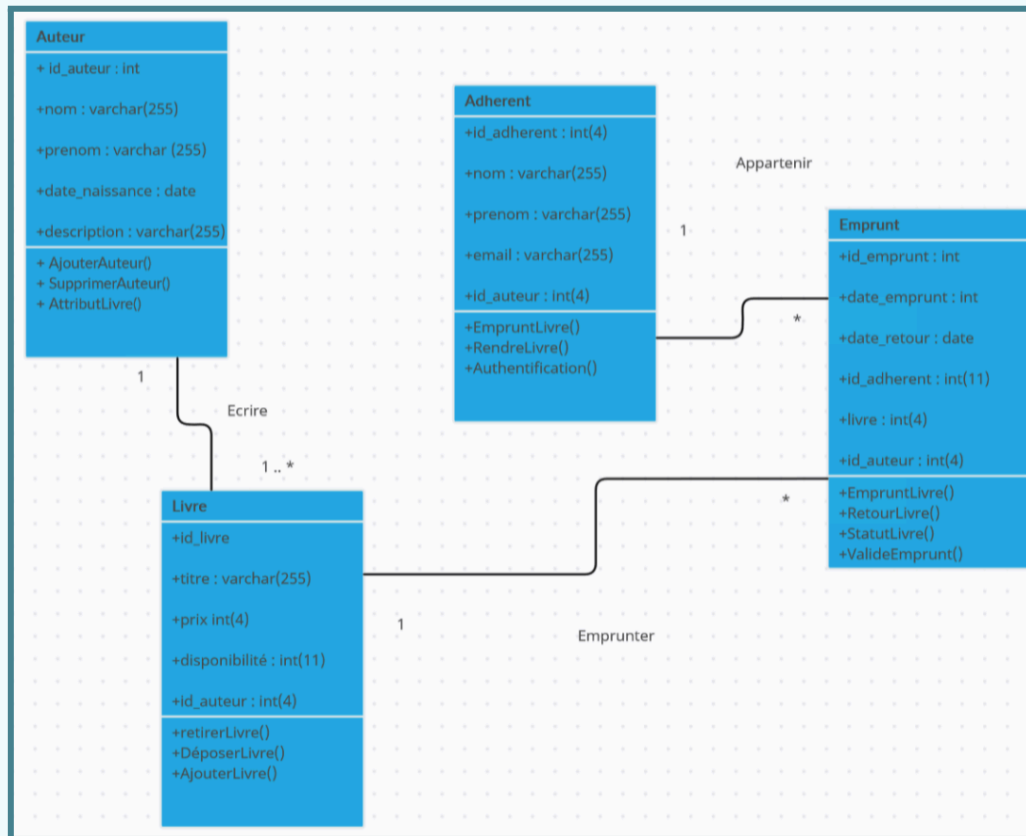
EMPRUNT (id_emprunt, date_emprunt, date_retour, id_adherent1 , livre, id_auteur, #id_livre, #id_adherent2)

LIVRE (id_livre, titre, prix, disponibilite, id_auteur 1#id_auteur 2)

3. Diagramme de classes

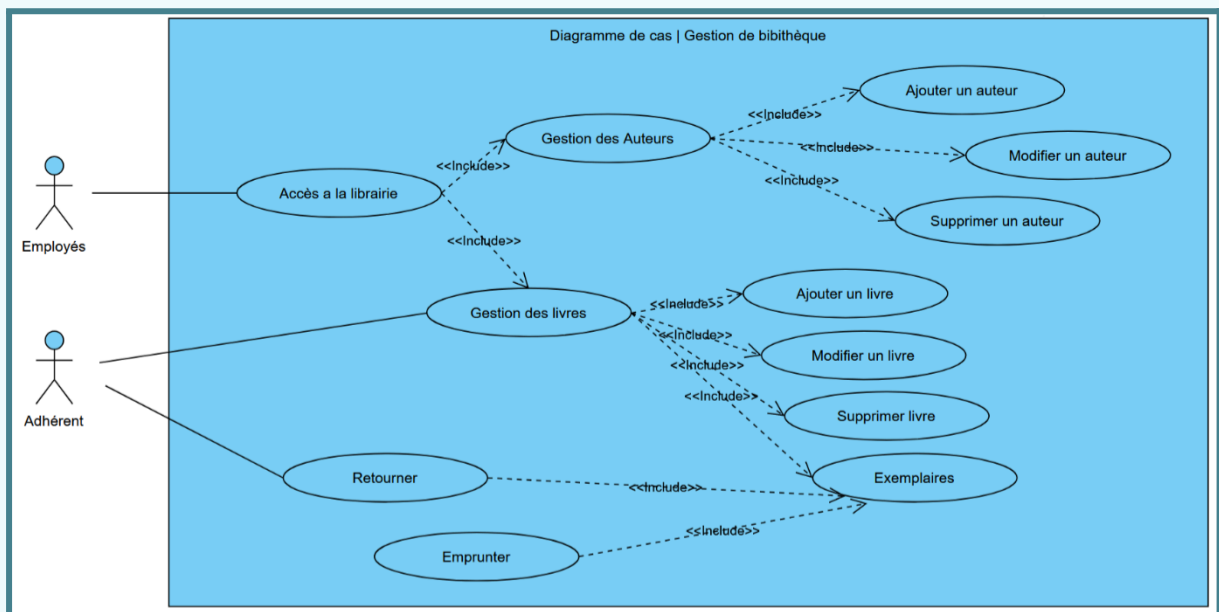
Figma





4. Diagramme de cas

figma



5. Liste des règles de gestion

1. Authentification et Accès Utilisateur

- **Règle 1** : Tout utilisateur doit être authentifié par un système de carte d'adhérent ou via une identification numérique avant d'utiliser la borne.
- **Règle 2** : Les utilisateurs non enregistrés peuvent parcourir le catalogue mais ne peuvent pas effectuer d'emprunts ou de réservations.

2. Gestion des Emprunts

- **Règle 3** : La durée maximale d'emprunt est fixée à 2 semaines, avec la possibilité de prolongation si aucune réservation n'est en attente sur le livre.
- **Règle 4** : Le nombre maximal de livres empruntables par l'adhérent est limité à 5 livres à la fois.

3. Réservations et Retours

- **Règle 5** : Les utilisateurs peuvent réserver des livres via la borne ou en ligne. Les livres réservés sont mis de côté pendant 7 jours.
- **Règle 6** : Les livres doivent être retournés à la borne. En cas de retard, des pénalités sont appliquées selon un barème préétabli.

4. Gestion des Stocks et Catalogue

- **Règle 7** : La borne doit être mise à jour en temps réel pour refléter la disponibilité des livres.
- **Règle 8** : Les suggestions d'achat de nouveaux titres par les utilisateurs peuvent être soumises via la borne.

5. Sécurité et Maintenance

- **Règle 9** : La borne effectue des diagnostics réguliers pour identifier d'éventuels problèmes de fonctionnement.
- **Règle 10** : Les informations des utilisateurs sont protégées conformément aux lois sur la protection des données.

6. Interaction avec le Personnel

- **Règle 11** : En cas de problème technique ou de demande spécifique, l'utilisateur peut solliciter l'aide du personnel via un bouton d'assistance sur la borne.

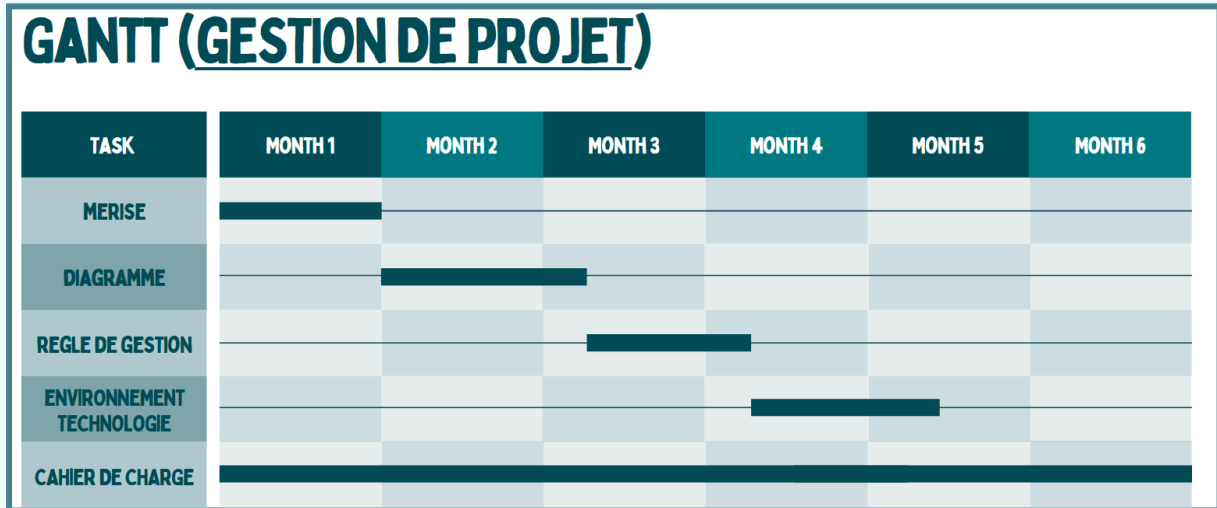
7. Mise à Jour du Logiciel

- **Règle 12** : Le logiciel de la borne est régulièrement mis à jour pour inclure de nouvelles fonctionnalités et améliorer la sécurité.



6. Diagramme de Gantt (gestion de projet)

Canva



7. Description de l'environnement technologique

7.1. Pare feu

- Utilisation d'un pare-feu matériel et/ou logiciel pour sécuriser le réseau du projet.
- Cisco ASA, Fortinet, ou un pare-feu logiciel comme pfSense.
- Protéger le réseau interne contre les accès non autorisés et les attaques externes.

7.2. Anti-virus

- Mise en place d'une solution antivirus pour protéger les systèmes contre les malwares et les virus.
- Symantec Endpoint Protection, Kaspersky Endpoint Security, ou Windows Defender pour les systèmes Windows.
- Assurer une protection en temps réel des postes de travail et serveurs.

7.3. OS version

- Choix des systèmes d'exploitation pour les serveurs et les postes de travail.
- Windows Server 2022 pour les serveurs, Windows 10 pour les postes clients.
- Fournir un environnement stable et sécurisé pour l'exécution des applications et services.



7.4. IDE Version

- Sélection des IDEs pour le développement des applications.
- Visual Studio 2022, IntelliJ IDEA, ou Eclipse.
- Offrir un environnement de développement efficace et intégré pour la programmation.

7.5. SGBD

- Utilisation d'un système de gestion de base de données pour stocker et gérer les données.
- MySQL, PostgreSQL, ou Microsoft SQL Server.
- Assurer la gestion, le stockage et la récupération sécurisée des données.

7.6. Navigateur

- Choix des navigateurs web pris en charge par les applications web du projet.
- Google Chrome, Mozilla Firefox, Microsoft Edge.
- Assurer la compatibilité et l'optimisation des applications web sur différents navigateurs.

7.7. Github

- Utilisation de GitHub pour le contrôle de version et la collaboration.
- Faciliter la gestion du code source, le suivi des modifications et la collaboration entre développeurs.

7.8. Les type de tests unitaires/fonctionnel

- Mise en place de procédures de tests pour assurer la qualité du logiciel.
- Tests unitaires avec JUnit ou NUnit, tests fonctionnels avec Selenium ou Cucumber.
- Valider la fiabilité, la performance et la conformité des fonctionnalités du logiciel.



Création d'un script pour création de la base de données (AP2)

Exemple de recherche d'un livre:

Maquette Interface accueil

Gérer la Bibliothèque

Gérer les Adhérents Gérer les Livres

Gérer les Auteurs Emprunter un Livre

Maquette Interface adherent

Gérer les Adhérents

Sélectionner ou ajouter un adhérent Mac John

Nom:

Prénom:

Email:

Ajouter Modifier Supprimer

Maquette Interface Livres

Gérer les Livres

Sélectionner ou ajouter un livre Le Joueur d'échecs

Titre:

Prix:

Auteur: Dostoevsky Fyodor

Disponibilité:

Ajouter Modifier Supprimer

Maquette Interface Auteur

Gérer les Auteurs

Sélectionner ou ajouter un auteur Dostoevsky Fyodor

Nom:

Prénom:

Date de naissance:

Description:

Ajouter Modifier Supprimer

Maquette Interface Emprunt

Emprunter un livre

Sélectionner un adhérent: LIN CLEMENT

Sélectionner un livre: Le Joueur d'échecs

Détails du livre:

Emprunter



Développement JAVA-bibliothèque:

```

1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class LibraryManagementPage extends JFrame {
6      public LibraryManagementPage() {
7          super(title:"Gérer la Bibliothèque");
8
9          JPanel panel = new JPanel(new GridLayout(rows:2, cols:2, hgap:10, vgap:10));
10         panel.setBorder(BorderFactory.createEmptyBorder(top:20, left:20, bottom:20, right:20));
11
12         JButton gererAdherentsButton = new JButton(text:"Gérer les Adhérents");
13         gererAdherentsButton.addActionListener(new ActionListener() {
14             public void actionPerformed(ActionEvent e) {
15                 openManageAdherentWindow();
16             }
17         });
18         panel.add(gererAdherentsButton);
19
20         JButton gererLivresButton = new JButton(text:"Gérer les Livres");
21         gererLivresButton.addActionListener(new ActionListener() {
22             public void actionPerformed(ActionEvent e) {
23                 openManageBooksWindow();
24             }
25         });
26         panel.add(gererLivresButton);
27
28         JButton gererAuteursButton = new JButton(text:"Gérer les Auteurs");
29         gererAuteursButton.addActionListener(new ActionListener() {
30             public void actionPerformed(ActionEvent e) {
31                 openManageAuthorsWindow();
32             }
33         });
34         panel.add(gererAuteursButton);
35
36         JButton emprunterLivreButton = new JButton(text:"Emprunter un Livre");
37         emprunterLivreButton.addActionListener(new ActionListener() {
38             public void actionPerformed(ActionEvent e) {
39                 openEmpruntLivreWindow();
40             }

```

```

41         });
42         panel.add(emprunterLivreButton);
43
44         add(panel);
45         setLocationRelativeTo(c:null);
46         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
47         pack();
48         setVisible(b:true);
49     }
50
51     private void openManageAdherentWindow() {
52         new ManageAdherentWindow();
53     }
54
55     private void openManageBooksWindow() {
56         new ManageBooksWindow();
57     }
58
59     private void openManageAuthorsWindow() {
60         new ManageAuthorsWindow();
61     }
62
63     private void openEmpruntLivreWindow() {
64         new EmpruntLivreWindow();
65     }
66
67     Run | Debug
68     public static void main(String[] args) {
69         SwingUtilities.invokeLater(new Runnable() {
70             public void run() {
71                 new LibraryManagementPage();
72             }
73         });
74     }

```

